

# CS2032 - Cloud Computing (Ciclo 2025-1)

## Proyecto Final

Semana 11 a Semana 15 (Exposición en Semana 16)

---

ELABORADO POR: GERALDO COLCHADO

# Agenda

## Proyecto Final

1. Competencias a lograr
2. Enunciado y Rúbrica
3. Entregables y Plazo

# Competencias a lograr

Por el alumno al finalizar el proyecto final

---

## Competencia:

*4.1: Crea, selecciona, adapta y aplica técnicas, recursos y herramientas modernas para la práctica de la computación y comprende sus limitaciones. (nivel 3).*

# Agenda

## Proyecto Final

1. Competencias a lograr
2. Enunciado y Rúbrica
3. Entregables y Plazo

# Enunciado y Rúbrica

## Proyecto Final

---

- **Integrantes:** Máximo 5 personas por grupo
  - **Funcionalidad:** Búsqueda de Productos y Compra. Ingesta en Tiempo Real.
  - **Arquitectura:**
    - Multi-tenancy
  - **Rúbrica:**
    - BackEnd (6 puntos)
    - Ingesta en Tiempo Real - Ciencia de Datos (6 puntos)
    - FrontEnd (5 puntos)
    - Diagrama de Arquitectura Solución (1 punto)
    - Exposición presencial (2 puntos)
- Si no se presenta a la exposición presencial su evaluación será desaprobatoria (sobre nota 10 como máximo).

# Enunciado

## Backend (6 puntos)

- **Elegir un tipo de producto según Grupo Canvas: Cloud Computing (CS2032) - Laboratorio 1.XX - 2025 - 1 > Personas > Grupos:**

Grupo 1: Medicinas (Referencia: <https://inkafarma.pe/>)

Grupo 2: Abarrotes (Referencia: <https://www.plazavea.com.pe/>)

Grupo 3: Productos Electrónicos (Referencia: <https://www.amazon.com/>)

Grupo 4: Libros (Referencia: <https://www.iberolibrerias.com/>)

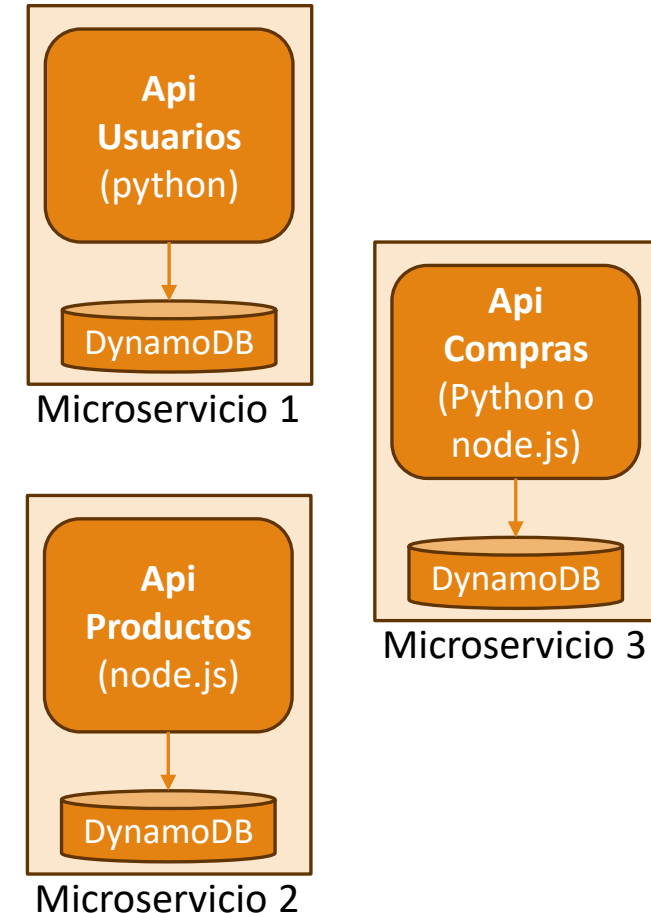
Grupo 5: Cursos Virtuales (Referencia: <https://www.udemy.com/>)

Grupo 6: Productos para mascotas (Referencia: <https://www.gopet.pe/>)

Grupo 7: Productos para bebés (Referencia: <https://www.maternelle.pe/>)

- **Microservicio Api Usuarios:** Multi-tenancy y Serverless en python con DynamoDB que permita crear usuario, login de usuario devolviendo token de acceso por 1 hora y validar token.
- **Microservicio Api Productos:** Multi-tenancy y Serverless y protegida con token en node.js con DynamoDB que permita ListarProductos (paginado), CrearProducto, BuscarProducto (Por Código), ModificarProducto y EliminarProducto.
- **Microservicio Api Compras:** Multi-tenancy y Serverless y protegida con token que permita registrar la compra de los productos por el usuario y listar todas las compras de un usuario. Utilice python o node.js con DynamoDB.

Nota: Para todos los microservicios debe automatizar su despliegue con framework serverless en 3 stages (dev, test y prod) incluyendo las tablas DynamoDB. Debe documentar las apis para visualizarlas en swagger-ui. Debe incluir enlaces a repositorios públicos de github con las fuentes.

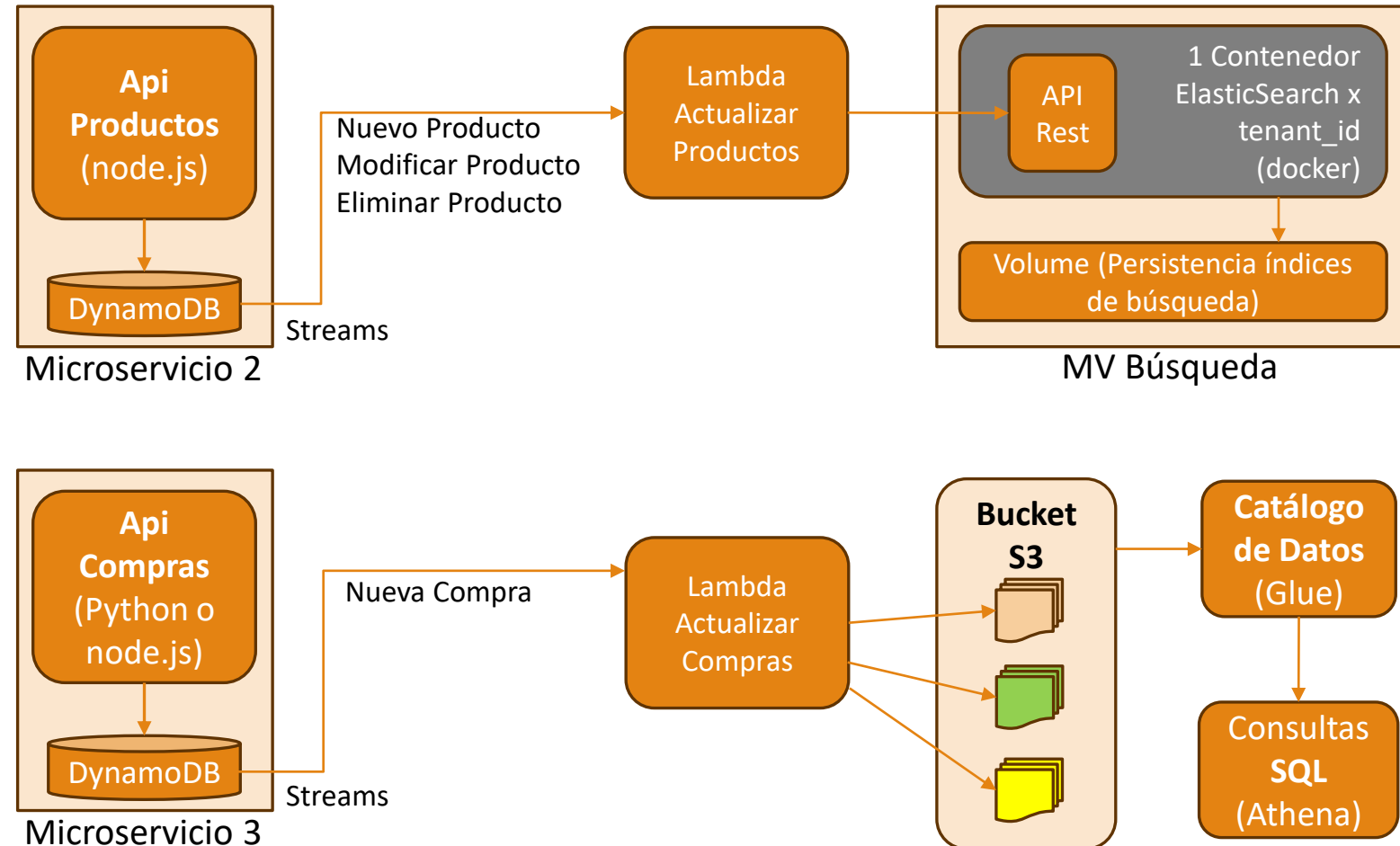


# Enunciado

## Ingesta en Tiempo Real (6 puntos)

### Ingesta en Tiempo Real

- **Change Data Capture (CDC):** Investigue y habilite la funcionalidad de DynamoDB Streams (\*) en las tablas DynamoDB de Api Productos y Api Compras para poder capturar en tiempo real los cambios.
- **MV Búsqueda:** Implemente una Máquina Virtual con 1 contenedor Elasticsearch (\*\*) por tenant\_id con su propio volumen y habilite sus apis rest.
- **Ingesta en Tiempo Real:** Implemente un Lambda Actualizar Productos que se conecte a DynamoDB Streams y actualice los productos en Elasticsearch. Implemente un Lambda Actualizar Compras que se conecte a DynamoDB Streams y actualice las compras como archivos CSV o json en S3 (Habilite un Catálogo de Datos y muestre evidencia de 3 queries SQL con Athena).
- Debe incluir enlaces a repositorios públicos de github con las fuentes de los lambdas de ingesta.



(\*) Referencia: [https://docs.aws.amazon.com/es\\_es/amazondynamodb/latest/developerguide/Streams.html](https://docs.aws.amazon.com/es_es/amazondynamodb/latest/developerguide/Streams.html)

(\*\*) Referencia: [https://hub.docker.com/\\_/elasticsearch](https://hub.docker.com/_/elasticsearch)

# Enunciado

## FrontEnd (5 puntos)

---

- Debe implementar una **página web Multi-tenancy** con opción de **crear usuario, login de usuario** y desplegarla en un **bucket S3** de AWS que tenga esta funcionalidad:
  - **Mantenimiento de Productos:** ListarProductos (paginado), CrearProducto, BuscarProducto (Por Código), ModificarProducto y EliminarProducto.
  - **Cuadro de texto para Búsqueda de Productos por su descripción:**
    - **Búsqueda Fuzzy (\*).** Ejemplo: Buscar “auriculares” en vez de “auriclares”.
    - **Búsqueda por Prefijo.** Ejemplo: Buscar todos los productos que inician con “auri”.
    - **Búsqueda con Autocompletado (\*\*).** Mientras el usuario escribe se muestra en línea los resultados de la búsqueda en la lista desplegable.
    - Elegir un producto de la lista o poder mostrar todos los productos encontrados en la web para su posterior elección, colocar la cantidad y adicionar al carrito de compras.
  - **Realizar compra:**
    - Visualizar todos los productos del carrito de compras con sus cantidades y registrar la compra.
    - Listar todas las compras del usuario.
- Puede usar el **framework web** del lado del cliente **de su preferencia**. Ejemplos: Javascript puro, Javascript con framework: react.js, angular.js, vue.js
- Debe incluir enlace a repositorio público de github con las fuentes.

(\*) <https://www.zewswweb.com/glosario/f/fuzzy-search-busqueda-difusa/>

(\*\*): <https://www.elastic.co/search-labs/blog/elasticsearch-autocomplete-search>



# Enunciado

## Diagrama de Arquitectura de Solución (1 punto)

---

- Debe elaborar un Diagrama de Arquitectura de Solución que incluya y relacione todos los componentes de:
  - BackEnd
  - Ingesta en Tiempo Real - Ciencia de Datos
  - FrontEnd

# Agenda

## Proyecto Final

1. Competencias a lograr
2. Enunciado
3. Entregables y Plazo

# Entregables y Plazo

## Grupos de hasta 5 personas

---

Entregables	Plazo (Fin de Semana 15)
<ul style="list-style-type: none"><li><i>Informe en word o pdf con evidencia de todo lo solicitado.</i></li><li><i>Resumen en power point con todo lo solicitado.</i></li><li><i>Exposición presencial y demo durante la semana 16.</i></li></ul>	<i>Máximo hasta Domingo 6-Julio-2025 23:59 (En Canvas)</i>